

## Шаблоны унифицированного языка моделирования для проектирования информационных систем

*Наиболее востребованными инструментами разработки объектно-ориентированных информационных систем являются CASE-средства. Чаще всего они не поддерживают  $n$ -арных отношений ассоциации между классами. Показано, как при описании модели предметной области можно перейти от  $n$ -арных ассоциаций к бинарным, как можно уменьшить число объектов информационной системы и как учитывать ограничения, накладываемые на объекты. Ситуации, которые позволяют осуществлять такого рода преобразования, представлены в виде шаблонов проектирования.*

**Ключевые слова:** UML, OCL, шаблоны проектирования, отношение ассоциации,  $n$ -арная ассоциация, тернарная ассоциация, кратность

### ВВЕДЕНИЕ

Бесспорным лидером среди инструментов, служащих для описания структуры и функционирования информационных систем (ИС), является унифицированный язык моделирования UML<sup>1</sup> [1-3], возможности которого используются в большинстве объектно-ориентированных, и в частности в гибких технологиях разработки информационных систем. На этапе их проектирования наиболее востребованные средства UML – диаграммы классов, которые служат для описания модели программной системы, отражающей основные закономерности предметной области. Базовым в диаграммах классов является отношение ассоциации, которое описывает потенциально возможные отношения между объектами программной системы. В дальнейшем в процессе разработки модель программной системы в виде диаграммы классов и диаграмм других типов трансформируется в логическую модель базы данных и в код объектно-ориентированного приложения, причем значительная часть кода может генерироваться автоматически с помощью CASE-средств<sup>2</sup>.

Значительную помощь при разработке модели предметной области могут оказать шаблоны проектирования [4-6]. Создавая новые шаблоны проектирования и ориентируясь на использование CASE-средств, необходимо учитывать, что, во-первых,

обычно CASE-средства не имеют адекватных механизмов представления  $n$ -арных<sup>3</sup> отношений ассоциации [7] (их реализация в отличие от бинарных достаточно трудоемка), во-вторых, поддержка объектного языка ограничений OCL<sup>4</sup> [8] обеспечивается большинством CASE-средств.

В настоящей статье будет показано, как учитывать семантику отношений ассоциации, как преобразовать тернарные ассоциации, возникающие между классами, в бинарные, как вводить дополнительные классы и бинарные отношения ассоциации с целью оптимизации модели и как с помощью объектного языка OCL учитывать ограничения, накладываемые на объекты, участвующие в отношениях ассоциации.

Рекомендации по созданию и преобразованию моделей ИС, повышающие эффективность их реализации [9-11], будут представлены в виде шаблонов проектирования.

### ЗАМЕНА ТЕРНАРНОЙ АССОЦИАЦИИ НА БИНАРНУЮ И КЛАСС-АССОЦИАЦИЮ

Рассмотрим несколько примеров. Первый пример: предположим, что в диаграмме классов имеется тернарная ассоциация, т. е. ассоциация, в которой участвуют три объекта. Например, возьмем объекты, принадлежащие трем разным классам: СТУДЕНТ, ПРЕДМЕТ и ЛЕКТОР (рис. 1). Определим кратности

<sup>1</sup>Unified Modeling Language – унифицированный язык моделирования, использующий несколько видов диаграмм для спецификации, конструирования и документирования программных систем.

<sup>2</sup>CASE-средство (Computer Aided Software Engineering) – программный инструмент, позволяющий автоматизировать процесс разработки информационных систем.

<sup>3</sup> $n$ -арное ( $n > 2$ ) отношение ассоциации – это отношение, в котором участвуют  $n$  объектов классов, входящих в ассоциацию.

<sup>4</sup>Object Constraint Language – объектный язык ограничений. Служит для определения накладываемых на объекты ограничений, которые не могут быть описаны на UML.

для классов, участвующих в этой ассоциации: СТУДЕНТ – (1..\*), ПРЕДМЕТ – (1..\*), ЛЕКТОР – (1).

Кратность ассоциации по отношению к классу ЛЕКТОР равна (1), поскольку любой фиксированной паре объектов из классов СТУДЕНТ и ПРЕДМЕТ соответствует только один объект класса ЛЕКТОР. Каждый ЛЕКТОР может вести занятия по одному предмету с несколькими студентами, поэтому кратность ассоциации по отношению к классу СТУДЕНТ равна (1..\*); один и тот же лектор может читать одному студенту сразу несколько курсов, поэтому кратность ассоциации по отношению к классу ПРЕДМЕТ равна (1..\*).

Заменяем тернарную ассоциацию на комбинацию бинарной ассоциации и класса-ассоциации. Доказательство справедливости такой замены достаточно простое. Соединим два класса с кратностями, отличными от (1), обычным отношением бинарной ассоциации. Тогда двум любым связанным объектам этих классов будет соответствовать ровно один объект

третьего класса, который мы можем без потери общности отнести к классу-ассоциации. Таким образом, класс-ассоциация заменит третий класс тернарного отношения. Более того, в этот класс мы можем дополнительно включить и атрибуты, относящиеся первоначально к тернарной ассоциации (рис. 2).

Это положение в принципе может быть распространено и на  $n$ -арные ассоциации и представлено в форме шаблона.

**Шаблон\_1:** Предположим, что в  $n$ -арной ассоциации имеется класс с кратностью (1). Тогда  $n$ -арную ассоциацию можно заменить на комбинацию  $(n-1)$ -арной ассоциации и класса-ассоциации.

Второй пример описывает случай, когда все объекты тернарной ассоциации принадлежат одному классу ЧЕЛОВЕК: имеется в виду отношение Отец – Мать – Ребенок (рис. 3). Здесь кратности таковы: ЧЕЛОВЕК (отец) – (1), ЧЕЛОВЕК (мать) – (1), ЧЕЛОВЕК (ребенок) – (\*).

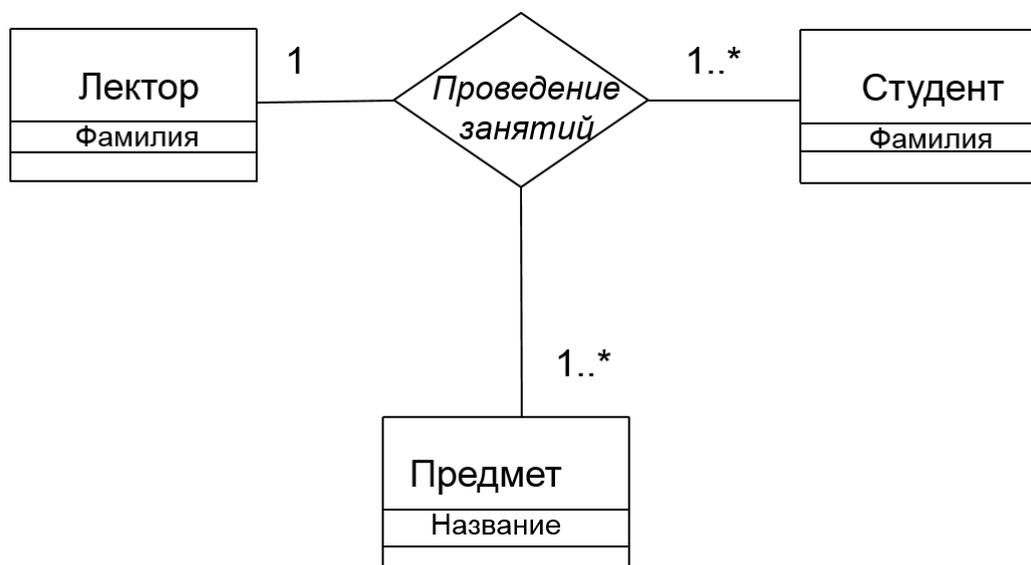


Рис. 1. Тернарная ассоциация.

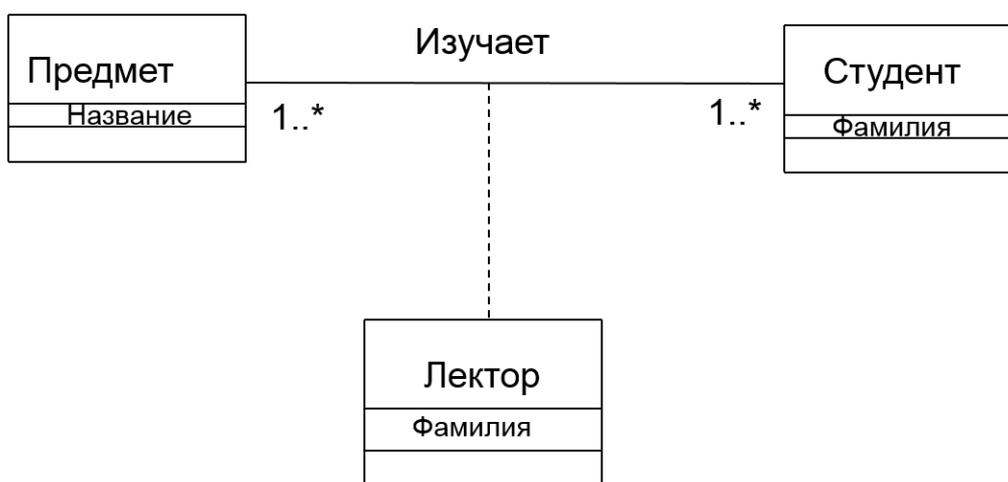


Рис. 2. Замена тернарной ассоциации на бинарную и класс-ассоциацию.

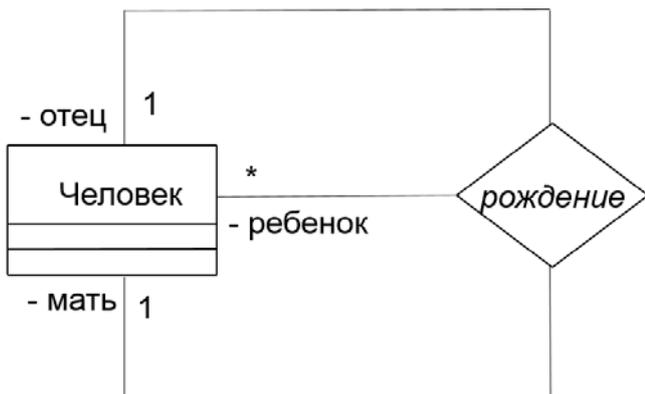


Рис. 3. Тернарная ассоциация между объектами одного класса

### ВВЕДЕНИЕ ДОПОЛНИТЕЛЬНЫХ КЛАССОВ

Возвратимся к первому примеру. Для тернарной ассоциации, как и для любой другой, могут существовать относящиеся к ней атрибуты. В конкретном случае это могут быть: время начала занятия, время окончания занятия, аудитория. Попробуем выделить из множества объектов одного класса подкласс, обладающий общностью по отношению к этим атрибутам. Например, из множества студентов формируются подмножества – группы или потоки, для которых проводятся аналогичные занятия в одно и тоже время. В этом случае возникает класс ГРУППА.

Тернарная ассоциация между классами СТУДЕНТ – ПРЕДМЕТ – ЛЕКТОР трансформируется в тернарную ассоциацию ГРУППА – ПРЕДМЕТ – ЛЕКТОР и бинарную СТУДЕНТ – ГРУППА. Определим кратности для классов, участвующих в новой тернарной

ассоциации: ГРУППА – (1..\*), ПРЕДМЕТ – (1..\*), ЛЕКТОР – (1).

В этом случае также, воспользовавшись Шаблон\_1, можно перейти к комбинации простой бинарной ассоциации и класса-ассоциации; тогда диаграмма классов примет уже другой вид (рис. 4). Преимущество такой диаграммы в том, что при реализации количество объектов – экземпляров класса-ассоциации ЛЕКТОР – будет существенно меньше.

Опираясь на изложенное выше, можно описать соответствующий шаблон проектирования следующим образом:

**Шаблон\_2.** Предположим, что для двух или более классов существует класс-ассоциация с одним или несколькими атрибутами. Если возможно объекты одного из классов разбить на подмножества, для которых значения атрибутов класса-ассоциации будут одинаковы, то следует ввести еще один класс и связать его отношением ассоциации с кратностями (1) и (\*) с первым классом.

Но все-таки избыточность в форме повторяющихся экземпляров класса-ассоциации ЛЕКТОР остается (в том случае, если лектор будет вести занятия в нескольких группах по одному предмету). Введем еще один класс-ассоциацию – РАСПИСАНИЕ. Это будет именно класс, а не атрибут, поскольку в него могут входить уже упомянутые нами атрибуты: время начала занятия, время окончания занятия, аудитория. И уже с этим новым классом свяжем обычной ассоциацией класс ЛЕКТОР. У этого нового отношения ассоциации будут следующие кратности: (1) – у класса ЛЕКТОР и (\*) – у класса РАСПИСАНИЕ (рис. 5). Такой вариант позволит уменьшить количество экземпляров класса-ассоциации ЛЕКТОР. Здесь мы используем операцию, аналогичную операции нормализации из теории баз данных.

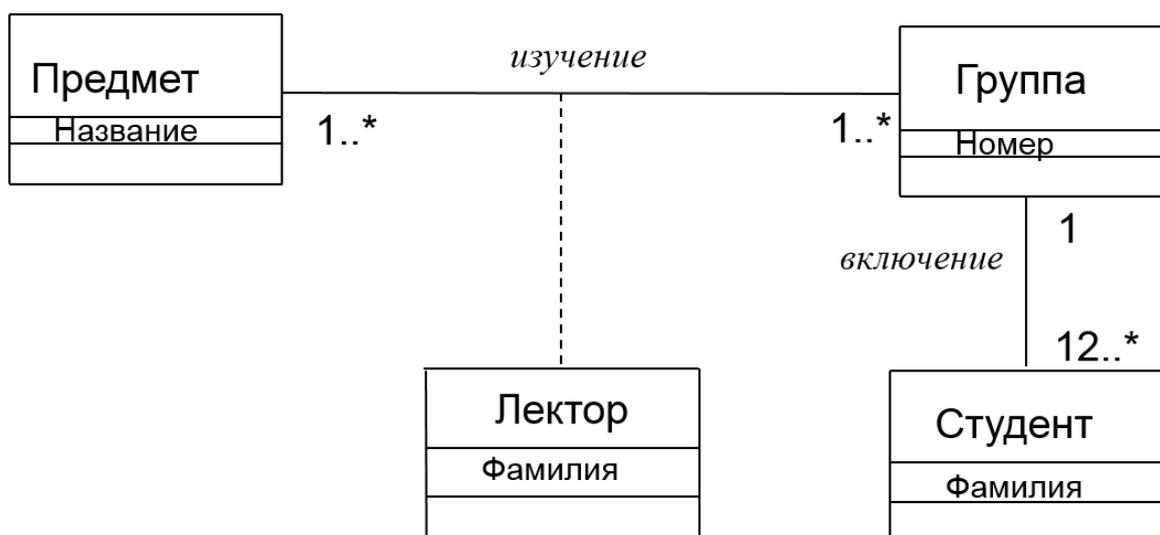


Рис. 4. Введение дополнительного класса ГРУППА

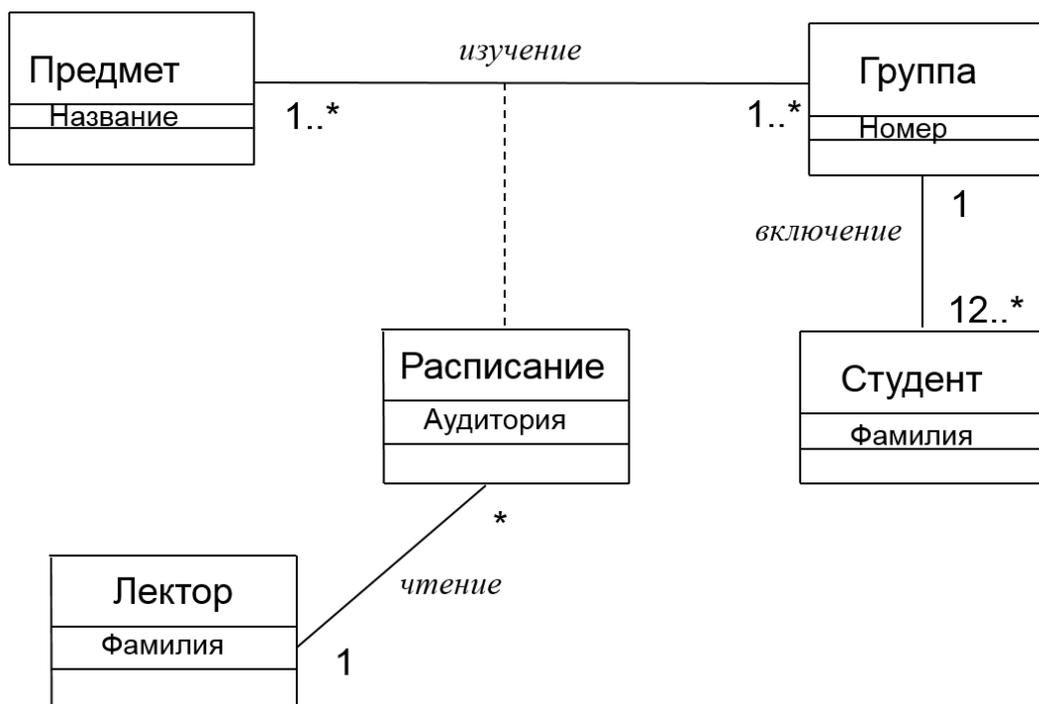


Рис. 5. Введение дополнительного класса-ассоциации – РАСПИСАНИЕ

### ЗАМЕНА *n*-АРНОЙ АССОЦИАЦИИ НА БИНАРНЫЕ

Опишем еще один, достаточно часто встречающийся тип *n*-арных отношений ассоциации, которые могут возникать между объектами классов. Предположим, что с одной стороны в отношении ассоциации участвует один объект одного класса, а с другой – произвольное число объектов второго класса. То есть отношение задается набором кортежей<sup>5</sup> переменной длины.

Приведем примеры таких отношений из реальных предметных областей. Пусть существуют два множества объектов, относящихся к заболеваниям, с одной стороны, и различным генотипическим признакам, с другой, из которых формируются классы: ЗАБОЛЕВАНИЕ и ГЕН\_ПРИЗНАК, соответственно.

Отношения между объектами этих классов можно кратко описать так: объект класса ЗАБОЛЕВАНИЕ может быть связан с любым количеством объектов ГЕН\_ПРИЗНАК, образуя кортежи разной длины. Кортежи в данном случае характеризуются еще и дополнительным атрибутом – вероятностью заболевания при наличии указанных генотипических признаков. Получается, что кортежи отношения, могут иметь, например, такой вид:

(31, ГП1), (31, ГП1, ГП2), (32, ГП2), (32, ГП3),  
(32, ГП1, ГП2, ГП3),

где 31, 32 – объекты класса ЗАБОЛЕВАНИЕ;

ГП1, ГП2, ГП3 – объекты класса ГЕН\_ПРИЗНАК.

Если отображать связи между объектами графически, то получится, что один объект класса ЗАБОЛЕВАНИЕ может быть связан с одним объектом ГЕН\_ПРИЗНАК более одного раза. Диаграмма объектов, соответствующая описанным выше в виде кортежей отношениям, имеет вид, представленный на рис. 6.

Очевидно, что средств для описания и задания такого рода отношений, когда в отношении ассоциации участвует один объект одного класса и произвольное число объектов другого класса, в языке UML нет. Но решить проблему можно путем ввода дополнительного класса ГРУППА\_ПРИЗНАКОВ. В случае возникает диаграмма классов, включающая кроме классов ЗАБОЛЕВАНИЕ, ГЕН\_ПРИЗНАК и ГРУППА\_ПРИЗНАКОВ еще и класс-ассоциацию между классами ЗАБОЛЕВАНИЕ и ГРУППА\_ПРИЗНАКОВ для хранения дополнительных атрибутов (рис. 7).

Тогда соответствующий шаблон проектирования может быть сформулирован так:

**Шаблон\_3.** Предположим, что в *n*-арной ассоциации участвует один объект одного класса и произвольное число объектов второго класса. В этом случае необходимо ввести новый класс для группирования объектов второго класса и связать его семантически различными отношениями бинарной ассоциации с первым и вторым классами.

<sup>5</sup> Кортеж – упорядоченная последовательность элементов, в данном случае объектов.

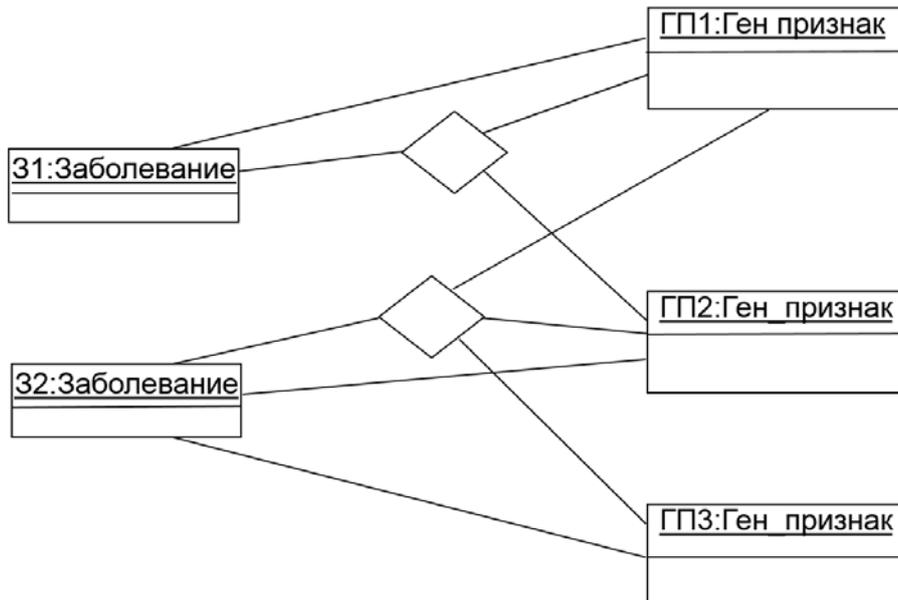


Рис. 6. Диаграмма объектов для предметной области – генетические заболевания

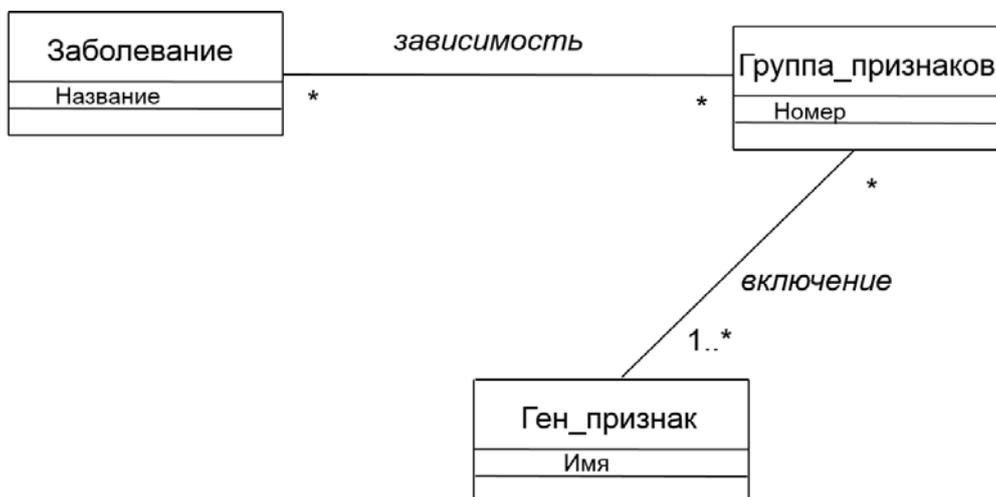


Рис. 7. Диаграмма классов для модели генетически обусловленных заболеваний

### ВЛИЯНИЕ ТОЧНОГО ОПРЕДЕЛЕНИЯ РОЛИ ОТНОШЕНИЯ АССОЦИАЦИИ НА СЕМАНТИКУ ОТНОШЕНИЯ

Предположим, что предметная область содержит данные о спортивных командах (название, бюджет) и играх (результат, место проведения) между ними. Будем считать, что каждая команда по одному разу играет со всеми остальными.

Пусть  $m$  – общее число команд. Тогда каждая команда проводит  $(m-1)$  матч со своими соперниками. Представим эту модель в виде диаграммы классов, которая будет включать класс КОМАНДА и отношение ассоциации ИГРА между объектами этого класса. Нас будут интересовать кратности для этого отношения. Очевидно, что кратность определяется числом игр и равна  $m-1$ . Причем кратность дублируется на обоих концах ассоциации. В этом случае число игр для каждой команды будет равно  $m-1$ , а само отношение будет симметричным (рис. 8).

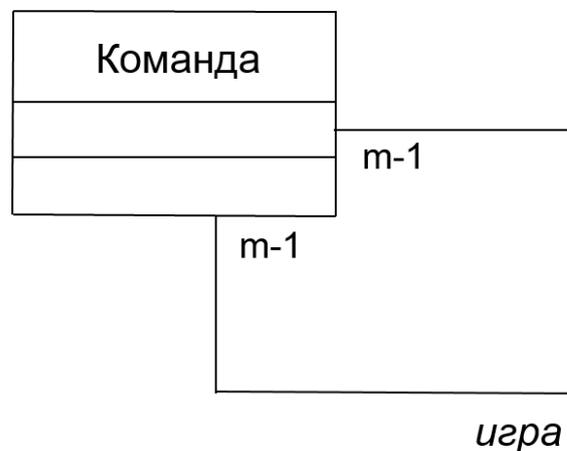


Рис. 8. Бинарное отношение ассоциации между объектами одного класса

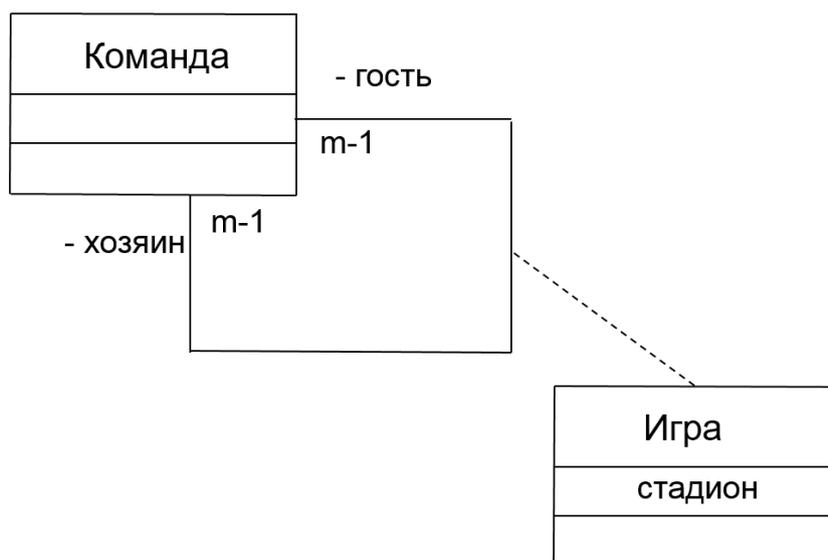


Рис. 9. Бинарная ассоциация между объектами одного класса с дополнительным классом-ассоциацией

Усложним задачу. Пусть каждая команда проводит  $2m-2$  матча и по два раза встречается с другими командами: на своем и чужом полях. Прежняя модель это не описывает, поскольку кратность отношения ассоциации не может превышать число объектов класса.

Для того, чтобы описать новую ситуацию, следует отказаться от симметричности отношения и ввести роли, назвав их, например, *хозяин* и *гость*. Тогда каждый объект класса КОМАНДА будет участвовать уже в  $2m-2$  отношениях: в половине случаев – в качестве хозяина, а в другой половине – в качестве гостя, хотя формально кратности отношения не изменились, т.е. фиксация роли позволила предать новый смысл отношению ассоциации, в котором участвуют объекты одного класса.

Для завершения формирования модели предметной области предположим, что у отношения ассоциации ИГРА имеются атрибуты: номер игры в календаре, дата встречи, стадион, число зрителей и т.п. Чтобы хранить значения этих атрибутов, необходимо ввести в модель предметной области класс-ассоциацию (рис. 9).

Обобщая изложенное, можно специфицировать шаблон 4 следующим образом:

**Шаблон 4. Точное определение ролей в отношении ассоциации может изменить семантику отношения, в частности, изменить число объектов, участвующих в ассоциации.**

Отметим, что отношение ассоциации всегда предусматривает упорядоченность объектов. Поясним это положение на более простом примере. Рассмотрим новое отношение ассоциации РАБОТА между классами КОМПАНИЯ и СОТРУДНИК. Определяя его, надо заранее указать, что первым в бинарной ассоциации стоит объект класса КОМПАНИЯ, а вторым – объект класса СОТРУДНИК, или наоборот. То же самое правило распространяется и на  $n$ -арные ассоциации.

Сложности возникают тогда, когда в отношении вступают объекты одного класса. Если отношение симметрично, то роли объектов в отношении одинаковы и их можно не определять. В таком случае порядок следования объектов не принципиален.

Если же отношение несимметрично, то необходимо определять роли объектов в отношении, т. е. вводить дополнительную семантическую информацию и фиксировать эти роли, устанавливая порядок следования объектов в отношении. Так при описании отношения РОДИТЕЛЬ – РЕБЕНОК роли надо указывать явно (рис. 10).

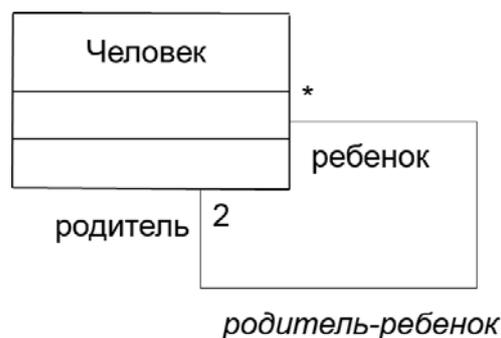


Рис. 10. Бинарная ассоциация между объектами одного класса с разными ролями объектов

## ДЕКОМПОЗИЦИЯ ТЕРНАРНОЙ АССОЦИАЦИИ НА ДВЕ БИНАРНЫХ АССОЦИАЦИИ

Обратимся снова к примеру с командами. Изменим условия задачи: будем считать, что каждые две команды (всего команд  $m$ ) встречаются между собой несколько раз. Тогда прежняя модель не пригодна для формализации новой ситуации. В принципе для описания предметной области в новой редакции достаточно всего одной тернарной ассоциации, включающей

классы ИГРА и дважды КОМАНДА (рис 11). Здесь  $n$  – число игр между двумя командами.

Интерес представляют значения кратностей: для команд они равны (1), а для игр – числу матчей между этими командами, в которых одна команда выступает исключительно либо в роли хозяина, либо в роли гостя.

Но нашу модель можно модифицировать и по-другому – заменив тернарную ассоциацию двумя бинарными (рис. 12). Новый класс ИГРА будет связан двумя отношениями ассоциации с классом КОМАНДА: первое отношение можно назвать ХОЗЯИН (имеется в виду хозяин поля в игре), второе – ГОСТЬ. Кратности со стороны класса КОМАНДА будут равны (1). Это однозначно определяет, какие команды участвуют в конкретной игре. Кратности со стороны класса ИГРА будут равны  $(m-1) \times n/2$ . Это выражение определяет общее число игр, которые каждая команда проводит в качестве гостя и хозяина соответственно.

Зададимся вопросом: эквивалентны ли модели, описываемые с помощью диаграмм классов, изображенных на рис. 11 и 12. Общее количество игр, проводимых каждой командой, и в том, и в другом случае равно  $(m-1) \times n$ . На своем и чужом полях также будет проведено равное число игр. Но, если в первом случае гарантируется, что каждая команда сыграет в роли хозяина со всеми другими ровно  $n/2$  матчей и столько же в роли гостя, то во втором случае это условие может не выполняться.

Для того чтобы подтвердить этот факт, рассмотрим конкретный пример: три команды играют турнир в два круга. Т. е.  $m=3$  и  $n=2$ , и каждая команда должна провести четыре матча, два из которых – в роли хозяина. Модель на рис. 12 допускает, что первая команда играет со второй оба матча дома, а с третьей оба матча в гостях, тогда как вторая команда играет с третьей оба матча дома, т. е. условие, когда каждая команда играет с другими равное число матчей в гостях и дома, не выполняется.

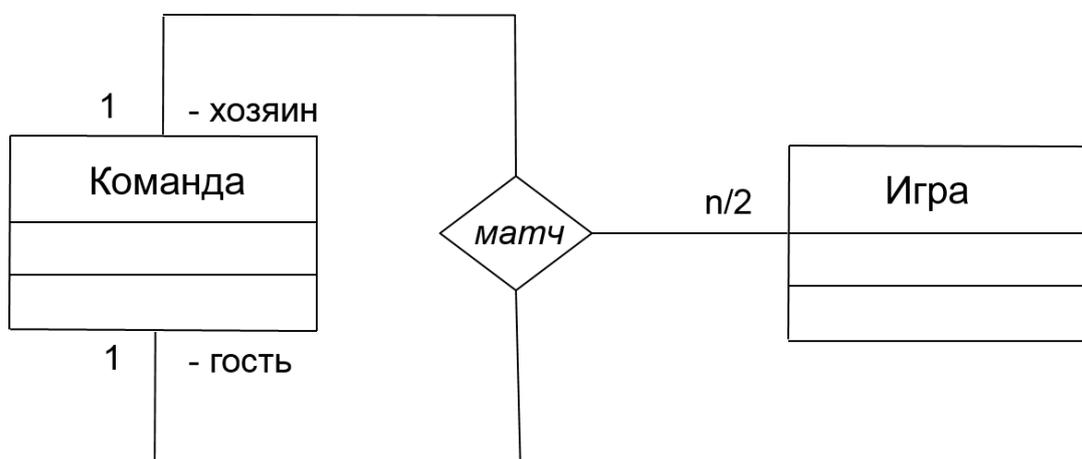


Рис. 11. Тернарная ассоциация, в которой два объекта принадлежат одному классу

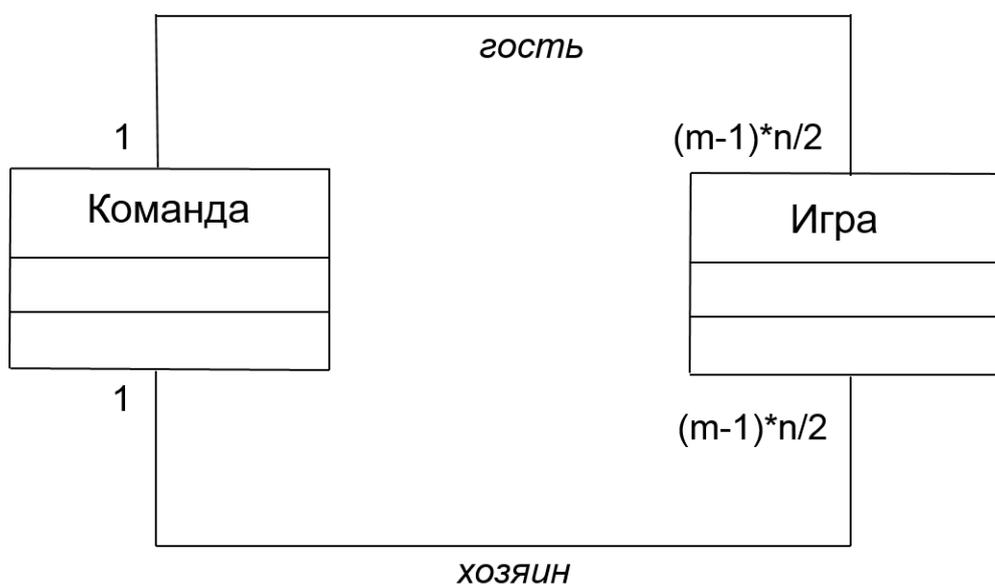


Рис. 12. Замена тернарной ассоциации двумя бинарными

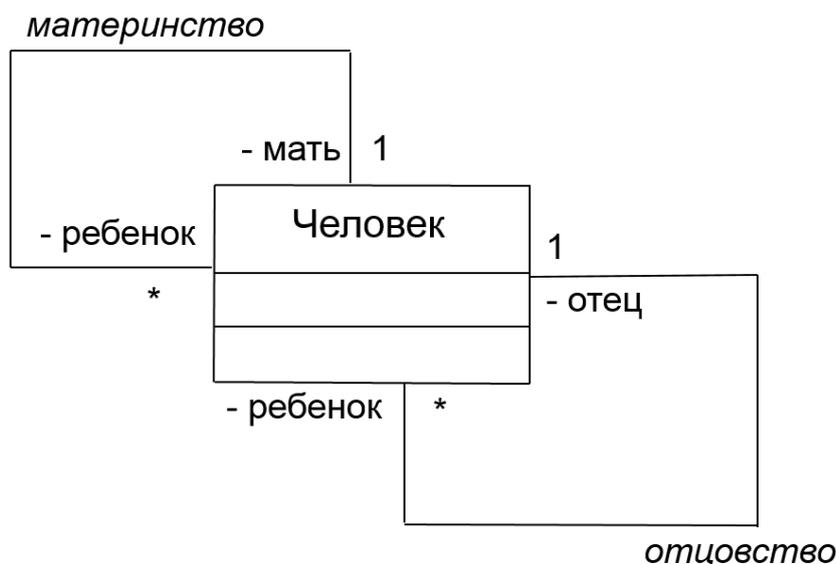


Рис. 13. Декомпозиция тернарной ассоциации между объектами одного класса

Рассмотрим более сложный случай, когда все объекты принадлежат одному классу. Самый очевидный пример представляет тернарное отношение МАТЬ – ОТЕЦ – РЕБЕНОК (рис. 3).

Эта ассоциация может быть декомпозирована на две обычных бинарных ассоциации (рис. 13). Это действительно так, поскольку по объекту ЧЕЛОВЕК в роли ребенка однозначно определяются его отец и мать.

Опираясь на написанное выше, можно представить правило декомпозиции в форме шаблона проектирования.

**Шаблон\_5. Тернарная ассоциация может быть декомпозирована на две бинарных ассоциации тогда, когда кратности отношения ассоциации для двух классов равны (1).**

Но, если в ассоциации принимают участие объекты одного класса, то при декомпозиции необходимо учитывать семантику ролей отношения. В противном случае модели могут оказаться не полностью эквивалентными.

### УЧЕТ ОГРАНИЧЕНИЙ НА ЗНАЧЕНИЯ ОБЪЕКТОВ, УЧАСТВУЮЩИХ В ОТНОШЕНИИ АССОЦИАЦИИ

Типичным является случай, когда объект не может вступать в отношение ассоциации с самим собой. Речь идет об отношении, в котором участвуют объекты одного класса. В основном это бинарные отношения, но встречаются и n-арные. С помощью стандартных средств языка UML такие ограничения описать невозможно. Тогда можно сделать следующий вывод:

**Шаблон\_6. Если бинарное отношение ассоциации специфицируется между объектами одного класса, то для того чтобы показать, что объект не может вступать в отношение с самим собой, необходимо использовать дополнительные ограничения.**

В частности, для ограничения множества объектов, вступающих в отношение ассоциации, отлично

подходит язык OCL [9]. Он позволяет либо определить те объекты, которые могут участвовать в отношении ассоциации, либо идентифицировать объекты, которые по законам предметной области не должны этого делать.

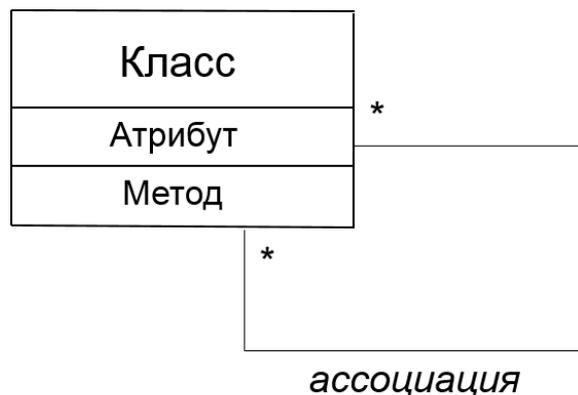


Рис. 14. Общий вид бинарной ассоциации между объектами одного класса

В общем случае (рис. 14), запрет будет выглядеть так:

*self.ассоциация -> exclude (self),*

где *self* представляет собой любой объект класса КЛАСС.

Чаще всего объект действительно не может вступать в бинарное отношение с самим собой, но возможны и исключения. Приведем пример. Пусть классом является ЧЕЛОВЕК, а отношением ассоциации, в котором участвуют объекты-люди, – ОТПРАВКА СООБЩЕНИЯ. Тогда вполне вероятно, что человек отправляет по электронной почте сообщение самому

себе, чтобы таким образом впоследствии получить доступ к содержимому сообщения с какого-то другого компьютера.

## ЗАКЛЮЧЕНИЕ

В настоящей статье описаны шесть шаблонов проектирования, которые могут быть использованы при разработке программных систем. Эти шаблоны показывают, как учитывать роли объектов в отношениях ассоциации, как преобразовывать  $n$ -арные ассоциации в бинарные с помощью введения новых классов и отношений, и как учитывать семантические ограничения, накладываемые законами предметной области. В реальных предметных областях перечисленные выше ситуации встречаются достаточно часто. И системным аналитикам необходимо находить выходы из таких ситуаций. Применение новых шаблонов, описанных в работе, дает возможность уже на стадии проектирования, во-первых, избавиться от сложных как для моделирования, так и для реализации  $n$ -арных ассоциаций, и во-вторых, уменьшить число возникающих при проектировании программной системы ошибок и неточностей, связанных с несоответствием модели и предметной области.

## СПИСОК ЛИТЕРАТУРЫ

1. Буч Г., Рамбо Д., Джекобсон А. UML. Руководство пользователя. – М.: ДМК, 2000.
2. Мацяшек Л. Анализ и проектирование информационных систем с помощью UML 2.0. – М.: Вильямс, 2016
3. Ларман К. Применение UML 2.0 и шаблонов проектирования. – М.: Вильямс, 2016
4. Gamma E., Johnson R., Helm R., Vlissides J. Design Patterns. Elements of Reusable Object-Oriented Software. – Addison-Wesley, 2001.

5. Sergievskiy M. N-ary Relations of association in class diagrams: design patterns // International Journal of Advanced Computer Science and Applications. – 2016. – Vol. 7, № 2. – P. 265-268.
6. Brown W., Malveau R., McCormick III H., Mowbray T. AntiPatterns, Refactoring software, architectures, and projects in crisis. – John Wiley & Sons, Inc, 1998.
7. Sergievskiy M., Kirpichnikova K. Optimizing UML class diagrams // ITM Web of Conferences, Volume 18, 7<sup>th</sup> Seminar on Industrial Control Systems: Analysis, Modeling and Computing (ICS 2018) Moscow, Russia, February 27-28, 2018. – URL: <https://www.itm-onferences.org/articles/itmconf/abs/2018/03/contents/contents.html>
8. Clark T., Warmer J. Object Modeling with the OCL. – Berlin: Springer, 2002.
9. Сергиевский М.В., Кирпичникова К.К. Валидация и оптимизация диаграмм классов UML // Cloud of Science. – 2018. – Т. 5. № 2 – С. 367-378.
10. Genova G., Llorens J., Martinez P. The meaning of multiplicity of n-ary associations in UML // Software System Modeling. – 2002. – № 1. – P. 86-97.
11. Olive A. Conceptual Modeling of Information Systems. – Berlin: Springer Science & Business Media, 2007.

*Материал поступил в редакцию 04.11.19.*

## Сведения об авторе

**СЕРГИЕВСКИЙ Максим Владимирович** – кандидат технических наук, доцент, доцент Национального Исследовательского Ядерного Университета МИФИ, Москва.  
e-mail: sermax@yandex.ru